



# Web căn bản



# **Buổi 2: HTML căn bản (Tiếp theo)**

# Nội dung buổi 2

- Làm đẹp thẻ HTML với thuộc tính style.
- Block-level và inline-level.
- Semantic HTML.
- HTML Form.
- Favicon.
- Giới thiệu về CSS.
- Hướng dẫn sử dụng git.

---

# Thuộc tính “style” – Làm đẹp cho giao diện

# Thuộc tính "style"

- Thuộc tính "style" dùng để để làm đẹp cho các thẻ HTML. Ví dụ, chọn màu cho chữ, màu cho nền của thẻ, kích thước của thẻ...
- Giá trị của thuộc tính "style" được viết bằng cú pháp của CSS. Bao gồm cặp một thuộc tính CSS đi với một giá trị. Giữa các cặp thuộc tính – giá trị này được ngăn cách bởi dấu ";".
- Ví dụ: **color: green; height: 120px**
- Trong đó: **color** và **height** là tên thuộc tính CSS, và **green**, **120px** là các giá trị của các thuộc tính tương ứng.

# Thuộc tính "style"

```
<p style="color: ■ yellow; background-color: ■ blueviolet">  
  This element has style  
</p>
```

**color: yellow** làm thay đổi  
màu chữ thành màu vàng

**background-color: yellow**  
làm thay đổi màu nền  
thành màu tím

This element has style

---

# Thẻ block và thẻ inline

# Thẻ block

- Thẻ block là các thẻ có chiều cao bằng kích thước nội dung của thẻ đó và chiều rộng chiếm 100% chiều rộng của thẻ bọc ngoài thẻ đó (hay thường gọi là thẻ cha).
- Theo mặc định, các thẻ này sẽ được xếp dọc lần lượt nối tiếp nhau theo như thứ tự thẻ hiện trong đoạn code.
- Các thẻ block mặc định: `body`, `main`, `header`, `footer`, `section`, `nav`, `aside`, `div`, `h1-h6`, `p`, `ul`, `ol`, `li`, ...



# Thẻ block

```
<body>
  <p style="background-color: ■yellow;">100% body width</p>

  <div>
    <p style="background-color: ■aquamarine;">100% div width</p>
  </div>

  <div style="width: 200px">
    <p style="background-color: ■aquamarine;">100% div width</p>
  </div>
</body>
```



100% body width

100% div width

100% div width

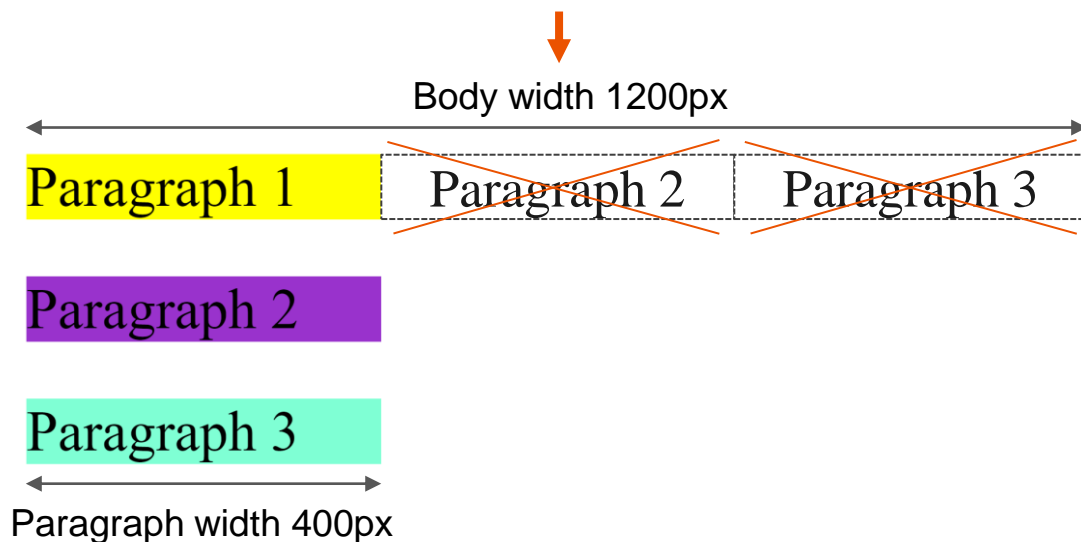
- Trong ví dụ bên, các thẻ <p> và thẻ <div> đều đều là thẻ block.
- Thẻ <p> đầu tiên có chiều rộng bằng thẻ <body>. Mặc định, thẻ <body> sẽ có chiều rộng bằng chiều rộng của cửa sổ trình duyệt.
- Thẻ <p> thứ hai có chiều rộng bằng thẻ <div> bao quanh nó. Tương tự, thẻ <div> cũng là thẻ block nên sẽ có chiều rộng bằng thẻ <body>.
- Thẻ <p> thứ ba có chiều rộng bằng thẻ <div> bao quanh nó. Tuy nhiên, thẻ <div> bọc quanh thẻ <p> này có quy định chiều rộng bằng 200px (do thuộc tính "style"). Nên chiều rộng lúc này của thẻ <p> thứ ba sẽ là 200px.

# Thẻ block

```
<body style="width: 1200px;">
  <div style="width: 400px">
    <p style="background-color: yellow;">Paragraph 1</p>
  </div>

  <div style="width: 400px">
    <p style="background-color: darkorchid;">Paragraph 2</p>
  </div>

  <div style="width: 400px">
    <p style="background-color: aquamarine;">Paragraph 3</p>
  </div>
</body>
```



- Các thẻ block sẽ luôn được xếp theo hàng dọc theo thứ tự từ trên xuống kể cả khi chiều rộng của thẻ nhỏ hơn chiều rộng của thẻ cha chứa nó.
- Như ví dụ ở bên, chiều rộng của thẻ `<body>` là 1200px, rộng gấp 3 lần chiều rộng của các thẻ `<div>` nằm trong nó. Tuy nhiên, các thẻ `<div>` này sẽ không dồn lên nằm cùng một hàng để lấp kín chiều rộng của thẻ `<body>`
- Lưu ý, đây là đặc tính cố định của thẻ block và không thể thay đổi. Trong trường hợp muốn tạo bố cục các thẻ nằm trên một hàng có thể sử dụng thẻ inline hoặc thuộc tính CSS display.

# Thẻ inline

- Thẻ inline là thẻ có kích thước chỉ bằng vừa đủ nội dung của nó (Không thể đặt chiều cao hay chiều rộng mới cho thẻ inline)
- Các thẻ inline khi đặt cạnh nhau sẽ nằm trên một hàng ngang theo thứ tự từ trái sang phải chứ không tạo một dòng mới xuống dưới như thẻ block.
- Chỉ khi tổng chiều rộng của các thẻ inline đặt nằm cạnh nhau lớn hơn chiều rộng của cửa sổ trình duyệt thì các thẻ inline mới sẽ được ngắt dòng nối tiếp xuống dưới (Logic tương tự như các trang gõ văn bản như Word...).
- Các thẻ inline mặc định: span, a, img, strong, em, button, ...

# The inline

```
<span>we</span>  
<span>-</span>  
<span>are</span>  
<span>-</span>  
<span>inline</span>
```



we-are-inline

---

# Bài thực hành số 1



# Bài thực hành số 1

- Bổ sung vào bài blog xây dựng từ các bài thực hành ở buổi 1.
- Sử dụng các thẻ inline `<a>` để tạo một thanh menu ở vị trí trên cùng của trang (page header). Nội dung của thanh menu có thể là: Trang chủ, Chủ đề, Bài viết mới, Liên hệ... (Có thể sử dụng thuộc tính CSS `padding-left`, `padding-right`, để tạo khoảng cách giữa các thẻ `<a>`. Ví dụ, `padding-left: 12px`)
- Sử dụng một số thuộc tính CSS đơn giản sau để làm đẹp cho các thẻ: `color`, `background-color`.

---

# Semantic HTML



# Semantic HTML là gì?

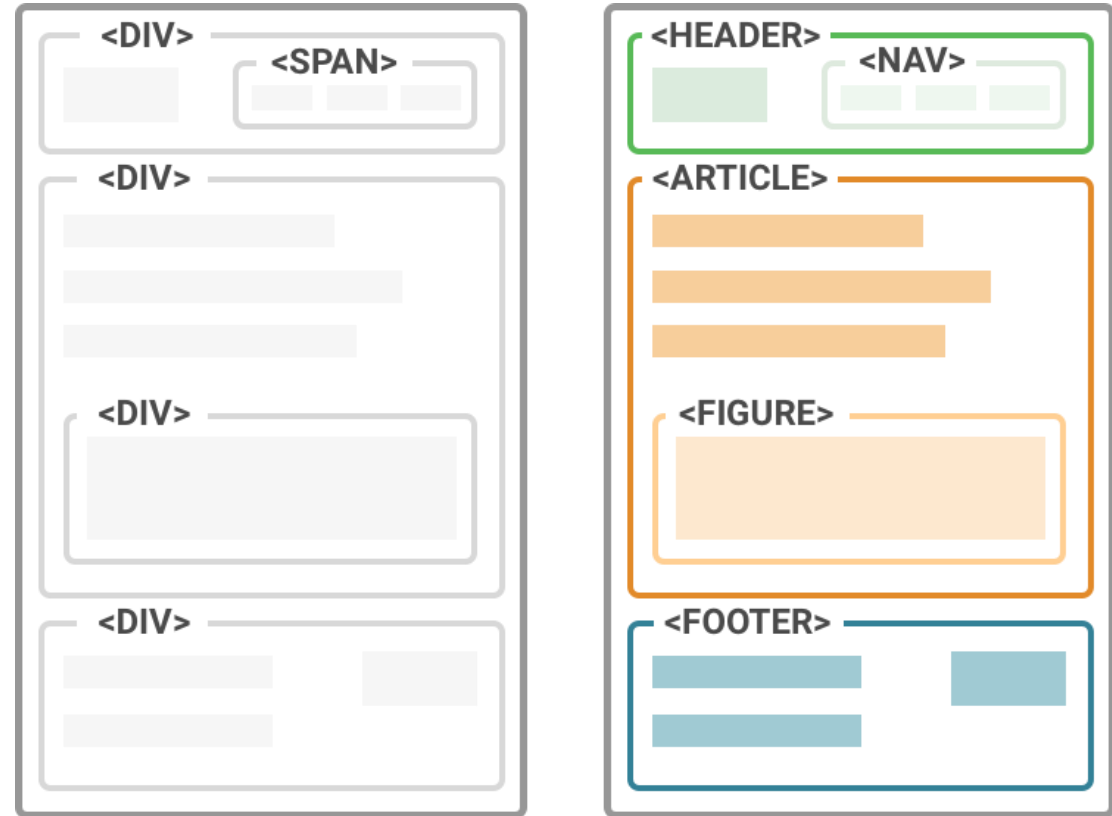
- Khái niệm Semantic HTML được giới thiệu ở phiên bản HTML 5.
- Các thẻ Semantic HTML có tác dụng mô tả chính xác nội dung mà nó chứa đối với cả trình duyệt và người phát triển.
- Ví dụ về các thẻ **non-semantic**: `div`, `span` – Chỉ có vai trò là thẻ chứa, không thể hiện thông tin gì về nội dung mà bên trong nó.
- Ví dụ về các thẻ **semantic**: `header`, `section`, `article`, `footer`... – Mô tả nội dung bên trong thẻ chứa.



# Thẻ Semantic HTML

## Một số thẻ Semantic HTML

- `<article>`
- `<aside>`
- `<details>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`



non-semantic vs semantic

# Vai trò của Semantic HTML

- Giúp các công cụ tìm kiếm đánh giá tốt hơn nội dung của trang web, tập trung vào các nội dung được cho là **important keywords** (Ví dụ như section, article, nav...).
- Hỗ trợ các thiết bị có tính năng đọc nội dung và định hướng người dùng bằng giọng nói (Ví dụ chức năng văn bản giọng nói hỗ trợ người dùng khiếm thị trên điện thoại thông minh).
- Thuận tiện cho lập trình viên trong quá trình thiết kế và thể hiện dữ liệu một cách có hệ thống trên trang web.

---

# Bài thực hành số 2



# Bài thực hành số 2

- Bổ sung vào bài blog xây dựng từ các bài thực hành ở buổi 1.
- Sử dụng các một số thẻ Semantic HTML (header, article, nav...) để bổ sung cho trang web theo chuẩn ý nghĩa nội dung.
- Tiếp theo nội dung blog đã xây dựng. Tạo thêm một phần nội dung mới với thẻ aside có tên "Bài viết tiếp theo" hoặc "Có thể bạn quan tâm"... Bên trong thẻ này gồm danh sách giới thiệu các bài blog khác, có bố cục gồm:
  - Ảnh minh họa bài blog.
  - Tiêu đề.
  - Một đoạn văn ngắn mô tả.

---

# HTML Form

# Thẻ form

- Thẻ form là một thẻ chứa một số thẻ đặc biệt như: input, label, textarea, có tác dụng để tạo thành một phiếu nhập thông tin, bảng hỏi dành cho người dùng để thu thập dữ liệu .Ví dụ như, thông tin đăng nhập (tên tài khoản, mật khẩu), thông tin yêu cầu tư vấn (họ tên, số điện thoại, email), thông tin thanh toán (đơn hàng, mã giảm giá)...
- Một số thẻ con thường sử dụng với thẻ form: input, label, textarea, select, option, button...

# Ví dụ mẫu sử dụng thẻ form

```
<form>
  <h2>Thông tin cá nhân</h2>
  <label for="full-name">Họ và tên</label>
  <input type="text" id="full-name">
  <br>
  <br>

  <label for="phone">Số điện thoại</label>
  <input type="text" id="phone">
  <br>
  <br>

  <button>Gửi thông tin</button>
</form>
```



## Thông tin cá nhân

Họ và tên

Số điện thoại

Gửi thông tin

# Thẻ label

- Thẻ label có tác dụng như tiêu đề nội dung của input mà nó đại diện.
- Thẻ sử dụng thuộc tính *for*, có nội dung trùng với id của thẻ input mà nó đại diện. Có tác dụng khi người dùng click chuột vào label, con trỏ soạn tạo văn bản sẽ tự động xuất hiện ở thẻ input tương ứng (Trong môi trường trình duyệt, đây gọi là sự kiện *focus*).

```
<label for="phone">Số điện thoại</label>  
<input type="text" id="phone">
```

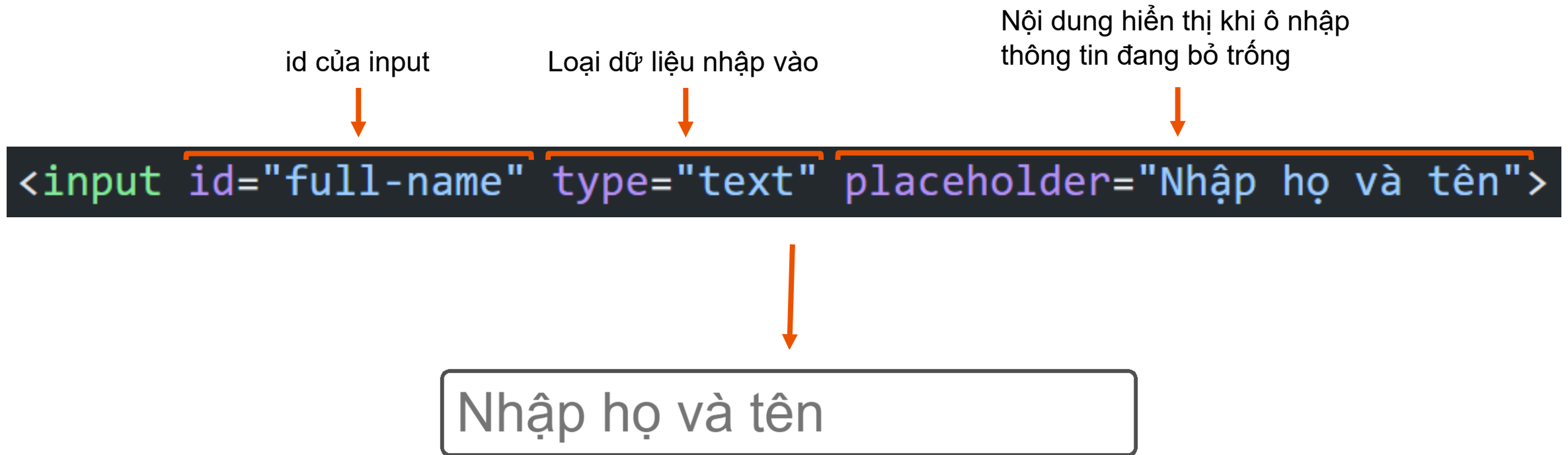


Số điện thoại



# Thẻ input – Dạng ô nhập văn bản

- Thẻ input sử dụng để tạo trường nhập thông tin trên trang web, có thể nhập dữ liệu văn bản, số, ô tích, file tải lên từ máy tính.
- Một số thuộc tính cơ bản của thẻ input:



# Thẻ input – Dạng checkbox

```
<h3>Các ngôn ngữ lập trình web front-end?</h3>
<input id="answer-1" type="checkbox" checked>
<label for="answer-1">HTML</label>

<input id="answer-2" type="checkbox" checked>
<label for="answer-2">CSS</label>

<input id="answer-3" type="checkbox">
<label for="answer-3">Python</label>
```

Các ngôn ngữ lập trình web front-end?

☒ HTML ☒ CSS ☐ Python

# Thẻ input – Dạng radio

```
<h3>Giới tính</h3>
<input name="gender" id="answer-1" type="radio">
<label for="answer-1">Nam</label>

<input name="gender" id="answer-2" type="radio">
<label for="answer-2">Nữ</label>

<input name="gender" id="answer-3" type="radio">
<label for="answer-3">Khác</label>
```

**Giới tính**

☒ Nam ☐ Nữ ☐ Khác

# Thẻ select, option

```
<label for="dev-type">Bạn là lập trình viên:</label>

<select id="dev-type">
  <option value="front-end">Front-end</option>
  <option value="back-end">Back-end</option>
  <option value="full-stack">Full-stack</option>
</select>
```



Bạn là lập trình viên: 

Front-end ▼

Front-end

Back-end

Full-stack

# Thẻ button

- Sử dụng để tạo thành phần dạng nút bấm trên trang web. Nội dung bên trong có thể là text hoặc chứa thẻ con khác.
- Thông thường thẻ button sẽ thường được trình duyệt định dạng sẵn một số style như background-color, border, hover effect.

```
<button>I'm a button</button>
```



I'm a button

# Thẻ button – Thuộc tính type.

- Khi thẻ button nằm trong 1 thẻ form, ta có thể đặt thuộc tính *type* cho thẻ button để thực hiện một số chức năng đặc biệt.
- Thuộc tính type của thẻ button nhận 3 giá trị:
  - “submit”: Gửi dữ liệu từ form. Đây là giá trị mặc định.
  - “button”: Đặt button trở về là một nút bấm thông thường.
  - “reset”: Xóa hết các thông tin đang nhập hoặc đặt lại giá trị mặc định của các thẻ input.

---

# Favicon

# Favicon

- Favicon là icon được hiển thị ở trên tab trình duyệt, ở vị trí bên trái cạnh tên của tab trình duyệt.



- Cách thêm favicon vào trang web: Ở tệp văn bản HTML, thêm thẻ link sau vào bên trong thẻ head.

```
<head>
  <!-- Các thẻ meta, title -->

  <link rel="icon" type="image/x-icon" href="src/img/favicon.png" />
</head>
```

Đường dẫn đến tệp ảnh trong thư mục dự án



---

# Bài thực hành số 3



# Bài thực hành số 3

- Tạo một form điền thông tin cá nhân.
- Thêm favicon cho trang web.
- Có sử dụng các input type như: text, email, checkbox, radio.
- Sử dụng các thẻ select, option.
- Có thể sử dụng thẻ `<br>` để các thành phần tạo thành các dòng mới.

---

# GitHub




# GitHub là gì?

- GitHub là một dịch vụ cung cấp kho lưu trữ mã nguồn Git dựa trên nền web cho các dự án phát triển phần mềm.
- Mỗi thư mục dự án của toàn bộ ứng dụng khi được lưu trữ trên GitHub sẽ được chia thành 1 **repository**, cho phép người dùng có thể cập nhật thường xuyên và cho nhiều thành viên khác tham gia làm việc cùng.
- Trang chủ GitHub: <https://github.com/>



# Cách tạo một Git Repository

- Bước 1: Tải và cài đặt Git (<https://git-scm.com/downloads>)
- Bước 2: Tạo tài khoản GitHub.
- Bước 3: Tại trang chủ. Chọn nút "New" → 
- Bước 4: Điền tên "Repository name" theo tên dự án, tên của ứng dụng.
- Bước 5: Chọn "Create repository". Hoàn tất tạo Repository trên GitHub.

# Tải lên GitHub thư mục dự án từ máy tính

- Bước 1: Mở cửa sổ Command Prompt và truy cập vào đường dẫn đến thư mục dự án bạn muốn tải lên GitHub (Mẹo: bạn có thể click chuột phải vào thư mục chứa các tệp dự án và chọn "Open in Terminal").
  - Bước 2: Nhập dòng lệnh sau:
- ```
git init
```
- Bước 3: Copy dòng lệnh tương tự với dòng lệnh sau từ repository mà bạn đã tạo và nhập vào Command Prompt.

```
git remote add origin https://github.com/HoangZoan/a.git
```

# Tải lên GitHub thư mục dự án từ máy tính

- Bước 4: Thêm các tệp và thư mục của dự án vào trạng thái *staging*.  
`git add .`
- Bước 5: Commit các tệp và thư mục ở trạng thái *staging*.  
`git commit -m 'noi dung message ma ban muon gui'`
- Bước 6: Đẩy các tệp và thư mục đã commit lên GitHub repository.  
`git push origin master`
- Bước 7: Hoàn tất, thư mục dự án đã được lưu trữ thành công trên GitHub.

---

# Hoàn thành Web căn bản – Buổi 2

Good job! 